DIMENSIONALITY REDUCTION

Yufan Cai, Yingzhe Luo, Feixiang XU, Xun Xu

Project 1 for Principle of Data Science, 2019 Spring Semester

• EXPERIMENT SETTING

– Experimental Environment

Python 3.5.2(Ananconda 4.2.0).

– Dataset

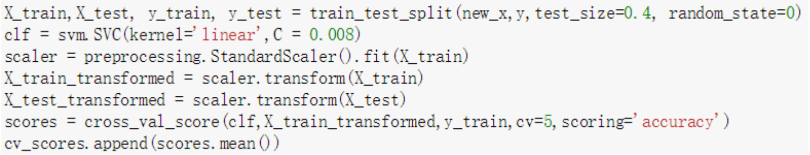
The dataset is downloaded from https://cvml.ist.ac.at/AwA2/, which consists of 37322 images of 50 animal classes.

– Feature

For each image, there are 2048 pre-extracted deep learning features.

– Training Testing Split

We split the images in each into 60 percents for training and 40 percent for testing and normalized the data. In order to avoid the problem of sample imbalance, we adopted stratified sampling and k-fold cross-validation. We also standardized the data.



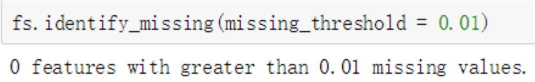
```
X_train,X_test, y_train, y_test = train_test_split(new_x,y,test_size=0.4, random_state=0)
clf = svm.SVC(kernel='linear',C = 0.008)
scaler = preprocessing.StandardScaler().fit(X_train)
X_train_transformed = scaler.transform(X_train)
X_test_transformed = scaler.transform(X_test)
scores = cross_val_score(clf,X_train_transformed,y_train,cv=5,scoring='accuracy')
cv_scores.append(scores.mean())
```

• DIMENSIONALITY REDUCTION METHODS

We use three methods to reduce dimensionality, i.e. feature selection, feature reduction, feature learning.

– Feature Selection

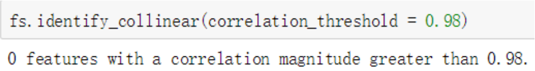
The first feature that needs to be eliminated is the feature with a large missing ratio. Then we find that no deep learning feature given by this data set actually has a ratio greater than 0.01.



```
fs.identify_missing(missing_threshold = 0.01)

0 features with greater than 0.01 missing values.
```

We then tried to identify features that were highly correlated with each other, which, due to their high variance and low interpretability, would lead to poor generalization of the test set data. Then we find that there is no feature with a correlation magnitude greater than 0.98.



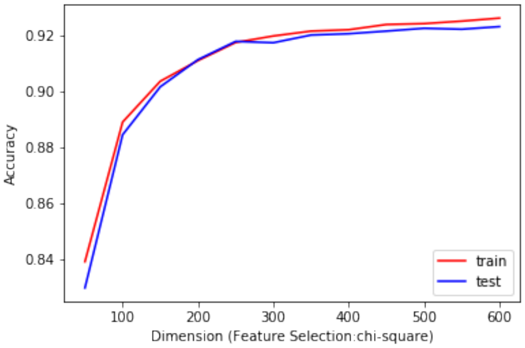
```
fs.identify_collinear(correlation_threshold = 0.98)

0 features with a correlation magnitude greater than 0.98.
```

It can be seen that the characteristics of deep learning given by the data set are relatively good. And then we do the univariate selection.

Univariate feature selection can test each feature, measure the relationship between the feature and the response variable, and throw away bad features according to the score. The method is simple, easy to operate and easy to understand, and it is usually effective for understanding data (but not necessarily effective for feature optimization and improving generalization ability). It is therefore recommended as a step in the preprocessing of feature selection.

Since it is a classification problem, we use chi-square test to select univariate features. We took the 50 to 450 most important features and tested them. The relationship between accuracy and dimension is shown below.



- **Feature Projection**
  Principal component analysis (pca) is one of the most classical data dimensionality reduction methods. Which assume that the data variables are linearly dependent and subject to a gaussian distribution. PCA is from the original space in order to find a set of orthogonal coordinate axes, among them, the first new axis option is one of the largest direction raw data variance, the second new axis is selected as the first axis orthogonal plane makes the variance of the largest and the third axis is with a 1, 2 axis orthogonal plane the variance of the biggest of all. And so on and so forth, we get n of these axes.

- **Feature Learning**
  For this part, we use t-SNE algorithm, which based on the probability distribution of random walk on the neighborhood graph and can find its structural relationship in the data. However, this algorithm has quadratic time and space complexity. Even if we pre-reduce the dimension to 200 with PCA, it is still a time-consuming task to carry out t-sne in tens of thousands of samples. At first, we tried to use this algorithm to visualize the samples in the three-dimensional space, and the results are shown in the figure below. Similar samples are marked with the same color.

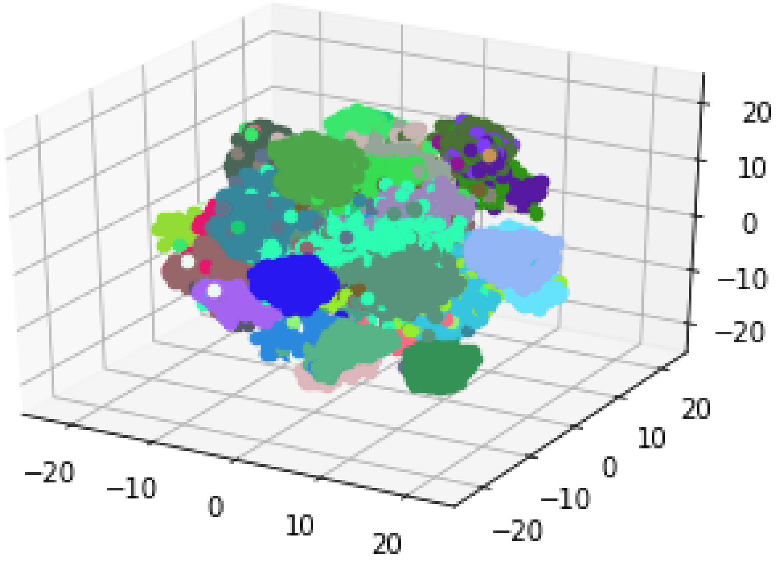


Figure 1: Visualization Trying

It can be seen that t-sne is indeed a very powerful algorithm for dimensionality reduction. If the dimension is reduced to 3D, t-sne can achieve nearly 90% accuracy, while PCA can only achieve about 25% accuracy and chi2 can only achieve about 15% accuracy.

- ## THE EXPERIMENTAL RESULTS

For this project, the hyper-parameter we determined is C in linear SVM. We use K-fold cross-validation within the training set to find C. Considering the complexity and effect of different dimensional reduction algorithms, we choose principal component analysis to explore the optimal dimension and optimal hyper-parameter.

  – **Optimal Hyper-parameter C**

  C is understood as adjusting the weight of preference of two indexes in the optimization direction (interval size, classification accuracy). The larger C is, the lower tolerance for classification errors is, and the easier it is to overfit. We use the linear SVM model with C equal to 1 to 30, respectively. Then we find the accuracy decreases almost monotonically as C increases. So let's change the interval for C to be $(0.1, 3)$.

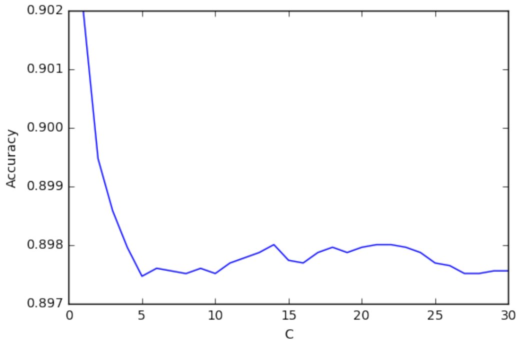


Figure 2: C from 1 to 30

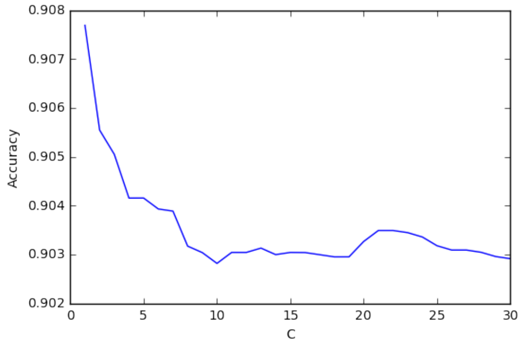


Figure 3: C from 0.1 to 3

  Unfortunately, it seems that we're going to make C a little bit smaller. Then we change the interval for C to be $(0.02, 0.4)$ and then $(0.002, 0.04)$. Finally, we find the optimal C which is 0.008. It can be seen that under the condition of C equals 0.008, the model is not easy to be over-fitted and the generalization ability is relatively strong.

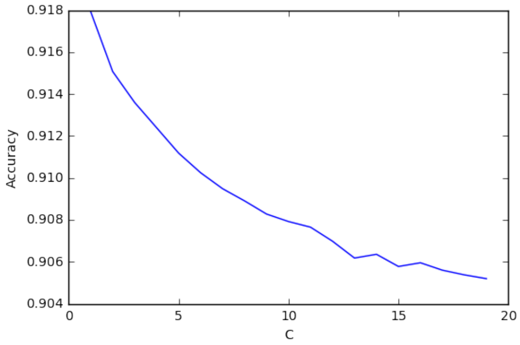


Figure 4: C from 0.02 to 0.4

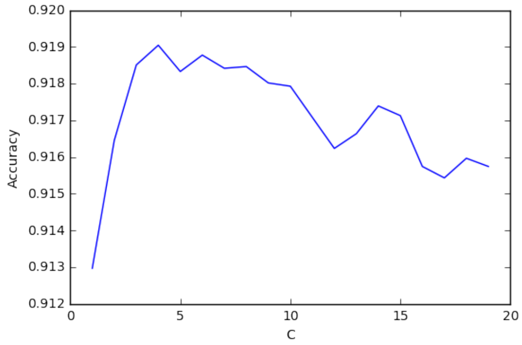


Figure 5: C from 0.002 to 0.04

  – **Optimal Dimension D**

  Because the accuracy of the feature selection algorithm(chi2) we implement increases monotonically with the increase of dimensions and the spatiotemporal complexity of the feature learning algorithm(t-SNE) is too high, we use the feature projection algorithm(PCA) to explore the optimal dimensions. In fact, under our experimental conditions, the accuracy of the algorithm reaches the maximum value around the dimension 182. Following is how we find this optimal dimension.

  The features of deep learning have a total of 2048 dimensions. First, we execute the SVM algorithm from 50 to 450 dimensions at an interval of 50 dimensions (using the best parameter C found before).

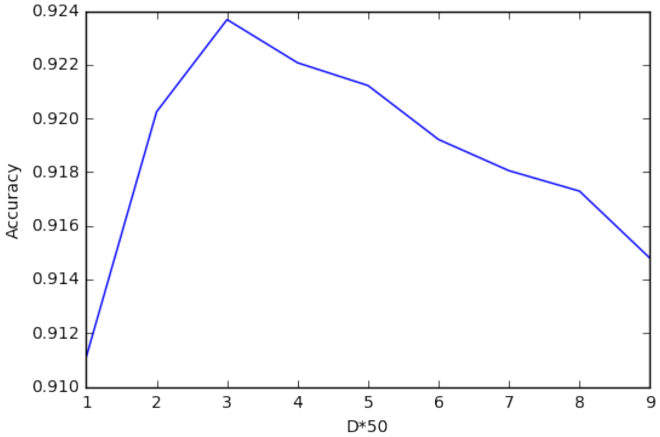


Figure 6: D from 50 to 450

So the optimal dimension is between 100 and 200.

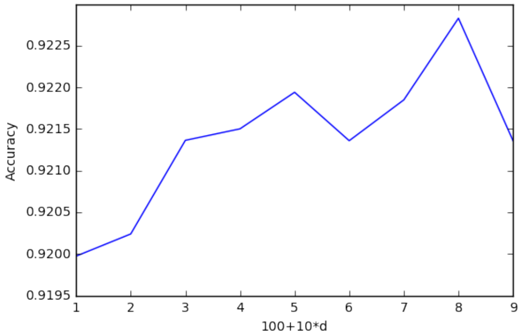


Figure 7: D from 100 to 190

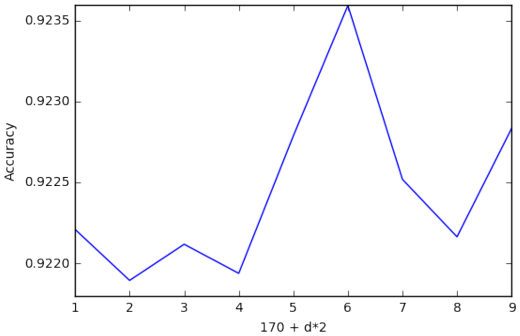


Figure 8: D from 170 to 188

Then we change the interval for D to be $(100, 190)$ and then $(170, 188)$.Then we find the optimal dimension which is 182.

- ## Experimental observations

  - ### Comparisons between algorithms

	Feature Selection(chi2)	Feature Projection(PCA)	Feature Learning(t-SNE)
Time cost	few minutes	few minutes	may be more than an hour
Characteristics	Monotonically increasing	Existence maximum	3D perform well
Suitable for	Data preprocessing	Optimal D finding	Visualization

  - ### About optimal dimension

For different algorithms, the optimal dimension has different definitions.
For feature selection algorithm adopted by the us, we didn't find the characteristics of the linear correlation as well as missing features. and the accuracy increases with the increase of the dimension. Considering the time and space complexity and accuracy increase quickly before they are slow, so it is difficult to define the best dimension, if only use the feature selection algorithm, we think the feature dimension is set to 200 to 300 is more appropriate.
For the feature learning algorithm, we only need to consider two and three dimensional cases. In this experiment, the distribution of all kinds of data and a few bad points can be clearly seen in both cases.

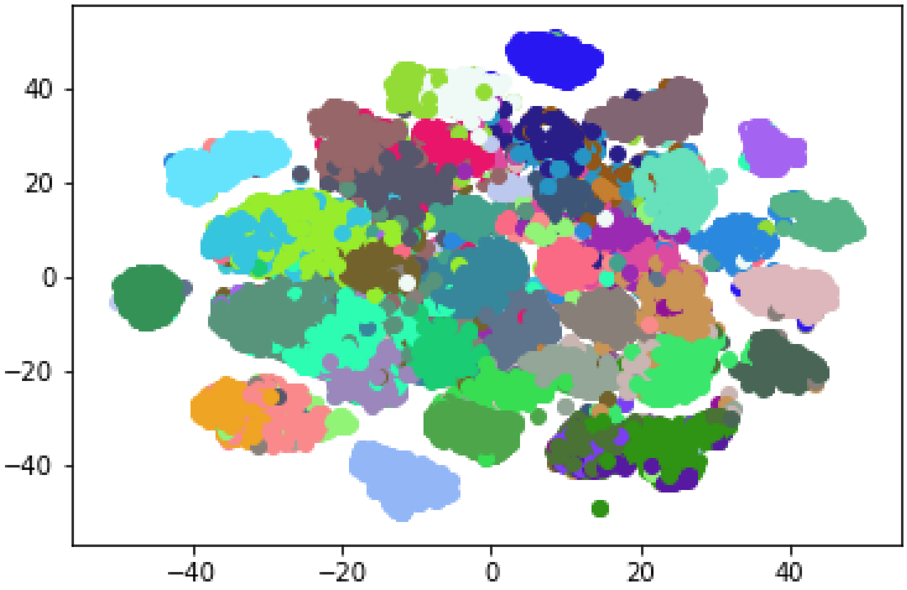


Figure 9: 2D Visualization using t-SNE

- **Thinking based on Visualization**

  According to our visualization results, we can clearly see that there are some misclassified points.The tolerance for these misclassified points reflects the choice of parameter C in linear SVM.After the experiment of our group, the final value of C is 0.008. It can be seen that our model has a relatively high tolerance for classification errors, so that it can have better generalization ability under the condition of ensuring no under-fitting.

  We think that if we can find these points that will be misclassified after data preprocessing, we can directly ignore them, and maybe we can get a more accurate, more generalized and more robust model.